

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

Claim 1 (Previously Presented): A method for performing computer memory initialization comprising:

upon starting computer memory initialization:

generating configuration data for a portion of memory;

saving the configuration data; and

restarting computer memory initialization;

upon restarting computer memory initialization:

copying the saved configuration data to initialize the portion of memory; and

using the portion of memory to execute instructions to initialize a remainder of memory.

Claim 2 (Original): A method as defined in claim 1 wherein:  
the portion of memory comprises at least one memory module; and  
the saving further comprises storing the configuration data in a nonvolatile memory device.

Claim 3 (Original): A method as defined in claim 2 wherein:  
the nonvolatile memory device is associated with the memory module.

Claim 4 (Original): A method as defined in claim 1 wherein:  
the restarting comprises a firmware initiated computer memory initialization; and  
the method further comprises:  
determining whether the computer memory initialization is initiated by firmware;

upon determining that the computer memory initialization is not initiated by firmware, performing the generating, saving and restarting; and  
upon determining that the computer memory initialization is initiated by firmware, performing the copying and using.

Claim 5 (Original): A method as defined in claim 1 further comprising:  
creating a software stack in the portion of memory; and  
using the software stack to execute stack-based instructions to initialize the remainder of memory.

Claim 6 (Original): A method as defined in claim 1 wherein:  
the portion of memory comprises a first memory controller and first memory associated with the first memory controller;  
the remainder of memory comprises a second memory controller and second memory associated with the second memory controller; and  
the copying further comprises copying the configuration data to the first memory controller to enable the first memory controller to use the first memory to execute the instructions to initialize the second memory controller and the second memory.

Claim 7 (Original): A method as defined in claim 6 wherein:  
the first memory associated with the first memory controller comprises first and second memory modules, the first memory module comprising an associated nonvolatile memory space; and  
the method further comprises:  
initializing the first memory controller to use the first memory module;  
using the first memory module to initialize the second memory module and generate the configuration data;  
saving the configuration data in the nonvolatile memory space associated with the first memory module; and  
copying the configuration data from the nonvolatile memory space associated with the first memory module to the first memory controller to enable the

first memory controller to use the first and second memory modules to execute the instructions to initialize the second memory controller and the second memory.

Claim 8 (Previously Presented): A method for initializing computer memory comprising:

resetting a computer system;

determining whether the reset is firmware initiated;

upon determining that the reset is firmware initiated, copying saved configuration data to initialize a portion of the computer memory, and using the portion of the computer memory to execute instructions to initialize a remainder of the computer memory; and

upon determining that the reset is not firmware initiated, generating and saving the configuration data and causing a firmware initiated reset of the computer system.

Claim 9 (Original): A method as defined in claim 8 further comprising:

creating a software stack in the portion of the computer memory.

Claim 10 (Original): A method as defined in claim 9 wherein:

the copying is performed using stackless instructions; and

the instructions executed to initialize the remainder of the computer memory are stack-based instructions.

Claim 11: Canceled.

Claim 12 (Previously Presented): A method as defined in claim 8 further comprising:

saving the configuration data in a nonvolatile memory.

Claim 13 (Original): A method as defined in claim 12 wherein:

the nonvolatile memory is associated with the portion of the computer memory.

Claim 14 (Original): A method for initializing computer memory, comprising:

resetting a computer system;

determining whether the reset is firmware initiated;

upon determining that the reset is not firmware initiated:

initializing a first memory controller and a first portion of a computer memory associated with the first memory controller using stackless code;

creating a first stack in the first portion of the computer memory associated with the first memory controller;

initializing the first memory controller and a second portion of the computer memory associated with the first memory controller using stack-based code;

saving configuration data for the first memory controller in a nonvolatile memory of the first portion of the computer memory associated with the first memory controller;

setting a firmware reset flag; and

resetting the computer system;

upon determining that the reset is firmware initiated according to the firmware reset flag:

copying the saved configuration data from the nonvolatile memory to the first memory controller;

creating a second stack in the computer memory associated with the first memory controller; and

initializing a second memory controller and a computer memory associated with the second memory controller using stack-based code.

Claim 15 (Original): A computer system, comprising:

a memory controller;

first and second computer memory associated with the memory controller;

stackless instructions; and

stack-based instructions;

and wherein:

under control of the stackless instructions the computer system

initializes the first computer memory for use by the memory controller; and  
under control of the stack-based instructions the computer system  
assembles configuration data which enables the memory controller to use the first  
and second computer memory.

Claim 16 (Previously Presented): A computer system, comprising:

a memory controller;

first and second memory modules;

a nonvolatile memory space; and

firmware, under control of which the computer system:

initializes the memory controller to use only the first memory  
module;

using only the first memory module, generates configuration  
data that enables the memory controller to use the first and second memory  
modules;

saves the configuration data in the nonvolatile memory space;

and

copies the configuration data from the nonvolatile memory  
space to the memory controller to initialize the memory controller to use the first and  
second memory modules.

Claim 17 (Original): A computer system as defined in claim 16 wherein the  
aforementioned memory controller is a first memory controller, further comprising:

a second memory controller;

a first computer memory comprising the first and second memory  
modules; and

a second computer memory;

and wherein:

under further control of the firmware the computer system:

initiates a reset after saving the configuration data in the  
nonvolatile memory space;

after the reset, configures the first memory controller to use the first computer memory upon copying the configuration data from the nonvolatile memory space to the memory controller; and

after the first memory controller is configured to use the first computer memory, configures the second memory controller to use the second computer memory.

Claim 18 (Previously Presented): A computer system, comprising:

a processor;

first and second memory controllers separate from the processor;

first and second computer memory associated with the first and second memory controllers, respectively;

a nonvolatile memory space; and

firmware under control of which the processor generates configuration data that enables the first memory controller to use the first computer memory, saves the configuration data in the nonvolatile memory space, copies the configuration data to the first memory controller to initialize the first memory controller to use the first computer memory, and uses the first computer memory to initialize the second memory controller to use the second computer memory.

Claim 19 (Original): A computer system as defined in claim 18 wherein:

the nonvolatile memory space is associated with the first computer memory.

Claim 20 (Original): A computer system as defined in claim 18 wherein:

under control of the firmware:

the computer system initiates a reset after saving the configuration data;

the computer system determines whether a current reset is firmware initiated;

if the current reset is firmware initiated, the computer system copies the configuration data to the first memory controller to initialize the first memory

controller to use the first computer memory, and uses the first computer memory to initialize the second memory controller to use the second computer memory; and  
if the current reset is not firmware initiated, the computer system generates the configuration data that enables the first memory controller to use the first computer memory, and saves the configuration data in the nonvolatile memory space.

Claim 21: Canceled.

Claim 22: Canceled.

Claim 23 (Previously Presented): A computer system comprising:  
first and second memory controllers;  
first and second computer memory associated with the first and second memory controllers, respectively;  
firmware comprising stackless instructions and stack-based instructions; and  
a nonvolatile memory space;  
and wherein:  
the first computer memory comprises first and second memory modules;  
under control of the stackless instructions the computer system initializes the first memory module for use by the first memory controller and creates a temporary software stack in the first memory module; and  
under control of the stack-based instructions the computer system uses the temporary software stack, generates configuration data that enables the first memory controller to use the first and second memory modules, saves the configuration data in the nonvolatile memory space, and executes a firmware initiated reset; and  
after the firmware initiated reset, under control of the stack-based instructions the computer system initializes the second memory controller to use the second computer memory.

Claim 24 (Original): A computer system as defined in claim 23 wherein:  
under control of the stackless instructions the computer system detects the firmware initiated reset, copies the configuration data to the first memory controller, and creates a permanent software stack in the first computer memory;  
and

under control of the stack-based instructions the computer system uses the permanent software stack and initializes the second computer memory for use by the second memory controller.

Claim 25 (Previously Presented): A computer system comprising:  
first and second computer memory;  
first and second means for controlling the first and second computer memory, respectively;  
a means, separate from the first and second controlling means, for generating configuration data that enables the first controlling means to use the first computer memory;  
a means for storing the configuration data during a firmware initiated reset;  
a means for executing the firmware initiated reset;  
a means, separate from the first and second controlling means, using stackless instructions upon a firmware initiated reset for copying the configuration data from the storing means to the first controlling means to use the first computer memory;  
a means for creating a software stack in the first computer memory;  
and  
a means, separate from the first and second controlling means, using stack-based instructions for initializing the second controlling means to use the second computer memory.